

# Belief Space Planning for Underwater Cooperative Localization

Jeffrey M. Walls, Stephen M. Chaves, Enric Galceran, and Ryan M. Eustice

**Abstract**—This paper reports on the inclusion of a probabilistic channel model within a cooperative localization planning framework. Underwater cooperative localization reduces positioning errors by sharing sensor data across a team of underwater vehicles. Relative range constraints between vehicles are measured by the one-way-travel-time of successfully received acoustic communication broadcasts. The quality of the navigation solution is intimately linked to the geometry of the network and, therefore, can benefit from planning informative relative trajectories. We cast this planning problem as an instance of belief space planning. In order to weight packet loss over the acoustic channel, we introduce a probabilistic channel model into the planning framework. We propose an optimization algorithm that allows us to plan open-loop control actions and, by extension, closed-loop parameterized trajectories.

## I. INTRODUCTION

Autonomous underwater vehicles (AUVs) integrate body-frame velocity, attitude, and pressure depth measurements for pose estimation. Bounded-error underwater localization remains a difficult challenge since horizontal position is not easily instrumented. Standard ground- or air-based position references such as GPS are unavailable underwater. Static acoustic beacon networks can provide an absolute position reference but do not scale to large vehicle networks and can be expensive to deploy [1].

Synchronous-clock cooperative localization enables teams of underwater vehicles to observe their relative range by measuring the one-way-travel-time (OWTT) of acoustic broadcasts. Each vehicle fuses local sensor data and relative range constraints to estimate its state. Navigation error can then be bounded by the collective sensing capability of the team. A variety of algorithms have been proposed to perform distributed inference for cooperative localization [2–5]. The overall quality of the navigation estimate, however, is strongly dependent on the network geometry.

A single OWTT-derived range measures the distance between a server vehicle’s time-of-launch (TOL) position and a client vehicle’s time-of-arrival (TOA) position. In this work, we assume that the client mission plan, i.e., desired trajectory, is available at planning time. Our goal is to compute a practical server trajectory that minimizes the predicted client uncertainty estimate through OWTT support.

Within OWTT cooperative localization there is an inherent trade-off between communication success and localization

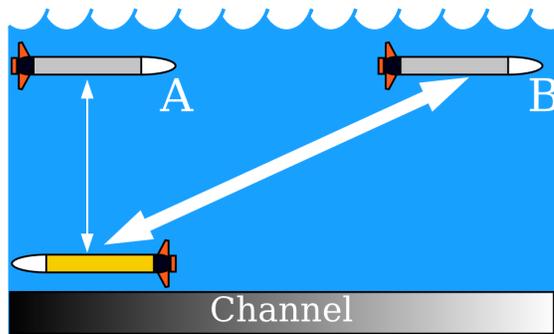


Fig. 1: Cooperative localization example: a server vehicle ‘A’ or ‘B’ can provide a relative range constraint (white arrows) to the client vehicle (yellow). ‘A’ is more likely to make the range observation as evidenced by the channel gradient (darker means higher probability of making a measurement). ‘B’, however, provides more information in the horizontal plane (as indicated by the thicker arrow). Our proposed planning framework balances these objectives.

accuracy—increasing relative range diminishes the probability of successful communication, but improves horizontal position estimation, as illustrated in Fig. 1. Belief space planning represents a principled approach to balancing standard path-planning objectives (e.g., minimum distance) with motion and sensing uncertainty. Resulting paths seek information in order to reduce state uncertainty and successfully meet a task-specific objective.

Gaussian belief space planning solves a more general partially observable Markov decision process (POMDP) problem over linear Gaussian models. Most belief space planning approaches, however, ignore that sensor measurements may not be obtained, for example, because of a lossy communication channel. We introduce a model of the underwater acoustic communication channel into belief space planning that allows a trajectory optimization algorithm to find server paths that provide useful navigation information and a higher fidelity communication channel (and hence provide informative relative range observations).

The specific contributions of this work include:

- We propose a belief space planning framework for server-client cooperative localization with a known client mission plan.
- We exploit a modified observation likelihood function that allows us to model randomness in obtaining measurements. We then employ this modified observation function to model the underwater acoustic channel.
- We propose an optimization algorithm for computing locally optimal open-loop control actions or path parameters that represent practical AUV trajectories.

Additionally, we provide a channel model that is representative of data collected during AUV field trials. We then use

\*This work was supported in part by the Office of Naval Research under award N00014-12-1-0092.

J. Walls, S. Chaves, E. Galceran, and R. Eustice are with the University of Michigan, Ann Arbor, Michigan 48109, USA  
{jmwalls, schaves, egalcera, eustice}@umich.edu.

the channel model to demonstrate our planning framework in several simulated scenarios.

## II. RELATED WORK

Path planning problems typically seek a feasible path from start to goal positions within a configuration space. When state is not known deterministically, planning can be formalized within a POMDP framework [6]. In this case, objective functions are defined over the space of potential distributions over state, called the belief space. Planning methods must cope with uncertainty in motion, sensing, and the surrounding environment. Due to the high dimensionality of the belief spaces of real-world problems, solutions to POMDP instances must typically be approximated [7].

Several prior approaches to belief space planning [8–10] attempted to solve belief space planning problems with sampling-based planners. They addressed uncertainty in sensing, motion, or both. These methods involve constructing candidate trajectories through the underlying state space by random sampling, then searching over candidate solutions. These algorithms find the best solution within the space of candidate trajectories but do not explicitly optimize the trajectory.

Recently, optimization motion planning has been applied to belief space planning [11–13]. This paper builds off of these methods in order to optimize over both open-loop control actions and parameterized trajectories. Several methods [11, 13] assume maximum-likelihood observations in order to achieve rapid replanning. Van den Berg et al. [12] drop the maximum-likelihood assumption at the cost of greater computational expense, but planning is performed off-line and a control policy is executed at run time.

The related area of active simultaneous localization and mapping (SLAM) research has focused on computing paths that lead to lower uncertainty maps. Stachniss et al. [14] presented an early particle-based approach to make informative decisions under uncertainty, integrating localization, mapping, and exploration. Sim and Roy [15] proposed a breadth-first search procedure for minimizing an A-optimal objective on vehicle uncertainty. Recently, Indelman et al. [16] merged optimization-based belief space planning within active SLAM. They additionally modeled unknown observation events (similar in spirit to the channel model presented here), but employed a simplification that results in a deterministic state covariance.

The multiple vehicle coordination problem is well studied [17], however, navigation for *localization* remains an open challenge. Martinez and Bullo [18] proposed a heuristic coordination algorithm for target-tracking with a range-only sensor. Charrow et al. [19] similarly presented a target-localization control policy using range-only sensors. The most relevant approaches to the context of underwater range-only localization include [20–22]. These prior works, however, do not include motion and sensing uncertainty and, in some cases, consider only discrete action spaces. Moreover, no prior work has modeled the communication channel, although several authors have imposed a heuristic maximum

communication range, e.g., Tan et al. [21]. We propose a principled method to incorporate channel behavior during planning.

## III. BELIEF SPACE PLANNING

The general planning problem we address in this work is an instance of a POMDP, since both sensing and motion are stochastic. We first state our problem instance within a Gaussian belief space (§III-A). Then, we modify the belief representation to handle nondeterministic observation events (§III-B). Finally, we develop a planning framework that leverages this modified belief representation (§III-C and §III-D).

### A. Gaussian belief space planning

Consider a system with state  $\mathbf{x}_k$  at time step  $k$ . The state transition model captures uncertainty in executing an action,

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k), \quad (1)$$

where  $\mathbf{u}_k$  is the control input, and  $\mathbf{w}_k \sim p(\mathbf{w}_k|\mathbf{x}_k, \mathbf{u}_k)$  is an independent noise disturbance with known density, assumed here to be additive zero-mean Gaussian with covariance matrix  $M_k$ . This transition model equivalently encodes the transition distribution  $p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k) = \mathcal{N}(f(\mathbf{x}_k, \mathbf{u}_k), 0), M_k)$ .

Observations are also random variables and modeled as

$$\mathbf{z}_k = h(\mathbf{x}_k, \mathbf{v}_k), \quad (2)$$

where  $\mathbf{v}_k \sim p(\mathbf{v}_k|\mathbf{x}_k)$  is an independent noise perturbation with known density. We assume that the observation noise is additive and drawn from a zero-mean Gaussian distribution with covariance matrix  $N_k$ . The observation model is then represented by the likelihood function  $p(\mathbf{z}_k|\mathbf{x}_k) = \mathcal{N}(h(\mathbf{x}_k), 0), N_k)$ .

The belief  $\mathbf{b}_k$  is the posterior distribution over state given all observations and control inputs up to time  $k$ , i.e.,

$$\mathbf{b}_k = p(\mathbf{x}_k|\mathbf{Z}_{1:k}, \mathbf{U}_{1:k-1}), \quad (3)$$

where  $\mathbf{Z}_{1:k} = \{\mathbf{z}_i\}_{i=1}^k$  is the set of observations and  $\mathbf{U}_{1:k-1} = \{\mathbf{u}_i\}_{i=1}^{k-1}$  is the set of all control inputs. In §III-B, we modify the observation and belief definitions to include unknown observation events.

In this work, we consider a general quadratic cost function over a trajectory comprised of the sum of a terminal cost and stage costs over the beliefs and controls through time-horizon  $L$

$$\min_{\mathbf{B}_{1:L}, \mathbf{U}_{1:L}} \mathbb{E} \left[ \mathbf{b}_L^\top \mathbf{Q}_L \mathbf{b}_L + \sum_{k=1}^{L-1} \mathbf{b}_k^\top \mathbf{Q}_k \mathbf{b}_k + \mathbf{u}_k^\top \mathbf{R}_k \mathbf{u}_k \right], \quad (4)$$

where  $\mathbf{B}_{1:L} = \{\mathbf{b}_k\}_{k=1}^L$  is the set of belief states that satisfy the relationship in (3), and the expectation is taken with respect to the random observations,  $\mathbf{Z}_{1:L}$ . The cost weight matrices  $\mathbf{Q}_k$  and  $\mathbf{R}_k$  are positive semi-definite and positive definite, respectively, and prescribe a preference for small deviations from a nominal belief and control. Quadratic (or approximately quadratic) cost functions are commonly

employed in belief space planning problems for the relative ease with which they can be minimized.

Computing the optimal solution to a cost function over belief states is generally intractable because of the high dimensionality of the space of all beliefs. Under the assumption of additive Gaussian noise and locally linear models, however, a belief is then approximately Gaussian and can be parameterized by the state mean  $\hat{\mathbf{x}}_k$  and vectorized covariance matrix  $\Sigma_k$ ,

$$\mathbf{b}_k = [\hat{\mathbf{x}}_k^\top, \text{vec}(\Sigma_k)^\top]^\top. \quad (5)$$

The  $\text{vec}(\cdot)$  operation exploits the symmetry in  $\Sigma_k$ . This Gaussian parameterization enables tractable optimization.

Our strategy closely follows previous methods for deriving an analytical belief evolution and optimizing the resulting trajectory [12, 23]. We proceed in two steps. First, we derive the belief dynamics, which define the belief transition distribution  $p(\mathbf{b}_{k+1}|\mathbf{b}_k, \mathbf{u}_k)$ . Second, we use the predictive belief dynamics to minimize a cost function over beliefs and controls (4). Our approach differs from the aforementioned works with the introduction of a channel model and uses an alternate optimization formulation that can be extended to consider parametric trajectories.

### B. Belief dynamics with uncertain channel state

Prior belief space planning methods assume that the set of future observation events is known. Whether or not an observation is made, however, is often not known before execution time. For example, the set of range observations that occur in an underwater acoustic network depends on the (unknown) set of successful acoustic transmissions.

To address this, we introduce a binary channel state variable,  $\gamma_k \sim \text{Bernoulli}(\lambda_k)$ , which models the event that an observation is received at the  $k$ th time index. Sinopoli et al. [24] introduced an equivalent model for studying convergence properties of the extended Kalman filter (EKF) error covariance in sensor networks with fading channels. We modify the observation model (2) by conditioning on the channel state variable ‘switch’

$$p(\mathbf{z}_k|\mathbf{x}_k, \gamma_k) = \mathcal{N}(h(\mathbf{x}_k, 0), \gamma_k \mathbf{N}_k + (1 - \gamma_k)\sigma^2 \mathbf{I}), \quad (6)$$

where we take the limit as  $\sigma^2 \rightarrow \infty$ . If the observation is received ( $\gamma_k = 1$ ), the revised model is exactly (2). When the observation is not received ( $\gamma_k = 0$ ), the observation has infinite uncertainty or, equivalently, zero information. Note that the observation model is still represented by a Gaussian distribution.

The belief now represents the posterior distribution over state given all channel states in addition to observations and control inputs

$$\mathbf{b}_k = p(\mathbf{x}_k | \mathbf{\Gamma}_{1:k}, \mathbf{Z}_{1:k}, \mathbf{U}_{1:k-1}), \quad (7)$$

where  $\mathbf{\Gamma}_{1:k} = \{\gamma_i\}_{i=1}^k$ . By leveraging the Markov property of the state transition model, the belief dynamics can be computed within a recursive Bayes filter

$$\begin{aligned} \mathbf{b}_{k+1} &= p(\mathbf{x}_{k+1} | \mathbf{\Gamma}_{1:k+1}, \mathbf{Z}_{1:k+1}, \mathbf{U}_{1:k}) \\ &= \eta p(\mathbf{z}_{k+1} | \mathbf{x}_{k+1}, \gamma_{k+1}) p(\mathbf{x}_{k+1} | \mathbf{x}_k, \mathbf{u}_k, \mathbf{\Gamma}_{1:k}, \mathbf{Z}_{1:k}), \end{aligned} \quad (8)$$

where  $\eta$  is a normalization constant and we assume that the set of channel states is independent of the state, i.e.,  $p(\gamma_i|\mathbf{x}_i) = p(\gamma_i)$ . While the channel state may certainly depend on the underlying state, we have assumed independence to simplify the belief dynamics and so that the belief is not informed by whether or not an observation is received—the channel model is only used at planning time and is uninformative during execution.

We track the evolution of belief states (8) with an EKF. We evaluate the state prediction and measurement update in the limit by substituting the observation model (6) into the Bayes filter (8). The resulting EKF update given the current belief closely follows the standard EKF update

$$\begin{aligned} \hat{\mathbf{x}}_{k+1} &= f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) + \gamma \mathbf{K}(\mathbf{z} - h(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0)) \\ \Sigma_{k+1} &= (\mathbf{I} - \gamma \mathbf{K} \mathbf{H}) \bar{\Sigma} \\ \mathbf{F} &= \frac{\partial}{\partial \mathbf{x}} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \\ \mathbf{H} &= \frac{\partial}{\partial \mathbf{x}} h(f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0), 0) \\ \bar{\Sigma} &= \mathbf{F} \Sigma_k \mathbf{F}^\top + \mathbf{M}_k \\ \mathbf{K} &= \bar{\Sigma} \mathbf{H}^\top (\mathbf{H} \bar{\Sigma} \mathbf{H}^\top + \mathbf{N}_k)^{-1}, \end{aligned} \quad (9)$$

where we have dropped the subscripts on  $\mathbf{z}_{k+1}$  and  $\gamma_{k+1}$  for brevity. Sinopoli et al. [24] present the equivalent update, although derived by first constructing the joint distribution over predicted state and observation and then conditioning on the observation. Within the update equations, the channel state variable multiplies the Kalman gain, resulting in an intuitive behavior—the standard Kalman update when the measurement is present, and an uncorrected process prediction otherwise.

The belief dynamics defined within the EKF update are random in the current observation and channel state,  $\mathbf{z}$  and  $\gamma$ , respectively. For tractable planning, we approximate the belief state transition with a Gaussian transition model

$$\mathbf{b}_{k+1} = g(\mathbf{b}_k, \mathbf{u}_k) + \mathbf{W}(\mathbf{b}_k, \mathbf{u}_k) \mathbf{w}_k, \quad (11)$$

where  $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{I})$ . Since  $\gamma$  is not Gaussian, we compute the above expression by moment matching, i.e., the first two moments of (11) are set to that of (9)–(10),

$$\begin{aligned} g(\mathbf{b}_k, \mathbf{u}_k) &= \mathbb{E}[\mathbf{b}_{k+1}] = \mathbb{E} \begin{bmatrix} \hat{\mathbf{x}}_{k+1} \\ \text{vec}(\Sigma_{k+1}) \end{bmatrix} \\ &= \begin{bmatrix} f(\hat{\mathbf{x}}_k, \mathbf{u}_k, 0) \\ \text{vec}(\bar{\Sigma} - \lambda \mathbf{K} \mathbf{H} \bar{\Sigma}) \end{bmatrix} \\ \mathbf{W}(\mathbf{b}_k, \mathbf{u}_k) &= \sqrt{\text{Var}[\mathbf{b}_{k+1}]} = \sqrt{\text{Var} \begin{bmatrix} \hat{\mathbf{x}}_{k+1} \\ \text{vec}(\Sigma_{k+1}) \end{bmatrix}} \\ &= \begin{bmatrix} \sqrt{\lambda \mathbf{K} \mathbf{H} \bar{\Sigma}} & \\ & \sqrt{\lambda(1-\lambda) \mathbf{v} \mathbf{v}^\top} \end{bmatrix} \\ \mathbf{v} &= \text{vec}(\mathbf{K} \mathbf{H} \bar{\Sigma}), \end{aligned} \quad (12)$$

where  $\mathbb{E}[\gamma] = \lambda$  and  $\sqrt{\cdot}$  is a suitable matrix square-root factor.

---

**Algorithm 1** Belief space trajectory optimization.

---

**Require:**  $\mathbf{b}_1, \bar{\mathbf{U}}^{(0)}$  {initial belief and nominal control}  
1: **while** not converged **do**  
2:  $\bar{\mathbf{b}}_{k+1}, A_k, B_k, W_k = \text{execute}(\bar{\mathbf{b}}_k^{(i)}, \bar{\mathbf{u}}_k^{(i)})$  for all  $k$   
3:  $G, J, W = \text{construct\_system}(\{\mathbf{b}_k, A_k, B_k, W_k\})$   
4:  $\delta\mathbf{U}^* = \text{solve\_objective}(Q, R, G, J, W)$   
5:  $\bar{\mathbf{U}}^{(i+1)} = \text{line\_search}(\bar{\mathbf{U}}^{(i)}, \delta\mathbf{U}^*)$   
6: **end while**  
7: **return**  $\bar{\mathbf{U}}^{(i)}$

---

Van den Berg et al. [12] present a similar belief dynamics formulation, although they have not considered random observation events. Under their model, the portion of the belief state vector corresponding to the state covariance matrix is deterministic, i.e., no random variables appear in the EKF covariance update when observation events are known. Here, the covariance depends on  $\gamma$ , and is therefore random. When an observation is guaranteed to be obtained, however,  $\lambda = 1$  and (11) is equivalent to the belief dynamics in [12].

The belief dynamics (11) represent the predictive transition of a belief state with random observation events. Kim and Eustice [25] and Indelman et al. [16] both considered the related problem of planning with unknown loop-closure events within active SLAM. Both parameterize Gaussian beliefs in the information (inverse covariance) form for which the measurement update is a simple additive step. Indelman et al. [16] (approximately) marginalize over  $\gamma$  in the posterior, as opposed to taking the expectation with respect to  $\gamma$ , resulting in a deterministic information (covariance) matrix. It is also worth noting that the belief dynamics are not invariant to belief parameterization. In the information form, as in [16, 25], the measurement information is scaled by  $\lambda$ , whereas, here, the Kalman gain is scaled instead.

### C. Belief space optimization motion planning

We use the belief dynamics defined in (11) to compute a locally optimal solution to the cost function defined by (4). Previously, van den Berg et al. [12] solved a similar objective using dynamic programming to obtain a control policy. Approaching the optimization as a batch process instead allows us to easily optimize over path parameters assuming a known path-following controller—desirable for planning practical trajectories for AUVs.

The cost function (4) can be expressed in batch as a function of the stacked vectors of beliefs and controls

$$\min_{\mathbf{B}, \mathbf{U}} E \left[ \mathbf{B}^\top \mathbf{Q} \mathbf{B} + \mathbf{U}^\top \mathbf{R} \mathbf{U} \right], \quad (14)$$

where we have dropped the subscripts on  $\mathbf{B}_{1:L}$  and  $\mathbf{U}_{1:L-1}$ ,  $\mathbf{Q} = \text{diag}(Q_1, \dots, Q_L)$ , and  $\mathbf{R} = \text{diag}(R_1, \dots, R_{L-1})$ . We take a sequential approach [26]—compute the set of beliefs satisfying the belief dynamics as a function of control actions, substitute into the objective function, and solve for the minimizing set of control inputs.

We first linearize each belief transition around a nominal trajectory  $\bar{\mathbf{B}}_{1:L}$  and control sequence  $\bar{\mathbf{U}}_{1:L-1}$

$$\delta\mathbf{b}_{k+1} \approx A_k \delta\mathbf{b}_k + B_k \delta\mathbf{u}_k + W_k \mathbf{w}_k, \quad (15)$$

where  $\delta\mathbf{b}_k = \mathbf{b}_k - \bar{\mathbf{b}}_k$ ,  $\delta\mathbf{u}_k = \mathbf{u}_k - \bar{\mathbf{u}}_k$ , and  $A_k$  and  $B_k$  are the Jacobians of  $g(\mathbf{b}_k, \mathbf{u}_k)$  with respect to  $\mathbf{b}_k$  and  $\mathbf{u}_k$ , respectively. Note that  $\bar{\mathbf{B}}_{1:L}$  is obtained by executing  $g(\cdot)$  along the nominal control trajectory given  $\mathbf{b}_1$ . We also linearize  $W(\mathbf{b}_k, \mathbf{u}_k)$  such that the  $i$ th column can be written

$$W_k^{(i)} \approx F_k^{(i)} \delta\mathbf{b}_k + G_k^{(i)} \delta\mathbf{u}_k + \bar{W}_k^{(i)} \quad (16)$$

and  $\bar{W}_k^{(i)} = W^{(i)}(\bar{\mathbf{b}}_k, \bar{\mathbf{u}}_k)$ . Note that  $W_k$  is a function of the channel model with parameter  $\lambda_k$ . In §V we discuss the form of  $\lambda_k$ , which depends on the state. During planning, we evaluate  $\lambda_k$  about the nominal trajectory.

We can write the stacked vectors of belief state and control deviations as

$$\delta\mathbf{B} = [\delta\mathbf{b}_1^\top, \dots, \delta\mathbf{b}_L^\top]^\top \quad (17)$$

$$\delta\mathbf{U} = [\delta\mathbf{u}_1^\top, \dots, \delta\mathbf{u}_{L-1}^\top]^\top. \quad (18)$$

The batch belief dynamics are then expressed by concatenating (15) over the time horizon

$$\begin{aligned} \delta\mathbf{B} &= A\delta\mathbf{B} + B\delta\mathbf{U} + W\mathbf{w} \\ \delta\mathbf{B} &= \underbrace{(\mathbf{I} - A)^{-1}B}_{\mathbf{G}} \delta\mathbf{U} + \underbrace{(\mathbf{I} - A)^{-1}W}_{\mathbf{J}} \mathbf{w}, \end{aligned} \quad (19)$$

where  $\mathbf{w}$  is the stacked vector of noise perturbations  $\mathbf{w}_k$  and the stacked Jacobians are defined

$$A = \begin{bmatrix} 0 & & & \\ A_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & 0 & A_{L-1} & 0 \end{bmatrix} \quad (20)$$

$$B = \begin{bmatrix} 0 & & & \\ B_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & 0 & B_{L-1} & \end{bmatrix} \quad (21)$$

$$W = \begin{bmatrix} 0 & & & \\ W_1 & 0 & & \\ 0 & \ddots & \ddots & \\ & 0 & W_{L-1} & \end{bmatrix}. \quad (22)$$

We substitute the batch belief dynamics (19) into the cost function (14) and evaluate the expectation

$$\begin{aligned} \min_{\delta\mathbf{U}} & [(G\delta\mathbf{U} + \bar{\mathbf{B}})^\top \mathbf{Q} (G\delta\mathbf{U} + \bar{\mathbf{B}}) + (\delta\mathbf{U} + \bar{\mathbf{U}})^\top \mathbf{R} (\delta\mathbf{U} + \bar{\mathbf{U}}) \\ & + \text{tr}(\mathbf{J}^\top \mathbf{Q} \mathbf{J})], \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min_{\delta\mathbf{U}} & [(G\delta\mathbf{U} + \bar{\mathbf{B}})^\top \mathbf{Q} (G\delta\mathbf{U} + \bar{\mathbf{B}}) + (\delta\mathbf{U} + \bar{\mathbf{U}})^\top \mathbf{R} (\delta\mathbf{U} + \bar{\mathbf{U}}) \\ & + \sum_i ((F^{(i)}G + G^{(i)})\delta\mathbf{U} + \bar{W}^{(i)})^\top (\mathbf{I} - A)^{-\top} \mathbf{Q} \\ & (\mathbf{I} - A)^{-1} ((F^{(i)}G + G^{(i)})\delta\mathbf{U} + \bar{W}^{(i)})]. \end{aligned}$$

This cost function is quadratic in the control update term,  $\delta\mathbf{U}$ , and can therefore be minimized by taking the derivative of the cost and setting to zero

$$\delta\mathbf{U}^* = -\mathbf{D}^{-1}\mathbf{E}, \quad (23)$$

where

$$\begin{aligned} \mathbf{D} &= \mathbf{G}^\top \mathbf{Q} \mathbf{G} + \mathbf{R} + \sum_i (\mathbf{F}^{(i)} \mathbf{G} + \mathbf{G}^{(i)})^\top \mathbf{C} (\mathbf{F}^{(i)} \mathbf{G} + \mathbf{G}^{(i)}) \\ \mathbf{E} &= \mathbf{G}^\top \mathbf{Q} \delta \mathbf{B} + \sum_i (\mathbf{F}^{(i)} \mathbf{G} + \mathbf{G}^{(i)})^\top \mathbf{C} \bar{\mathbf{W}}^{(i)} \\ \mathbf{C} &= (\mathbf{I} - \mathbf{A})^{-\top} \mathbf{Q} (\mathbf{I} - \mathbf{A})^{-1}. \end{aligned}$$

We leverage the sparsity structure of  $\mathbf{W}^{(i)}$  to efficiently compute the summation terms. Note that the Hessian is positive definite by construction because of the cost weight matrices. The updated control vector is computed by adding the update to the control at the current iteration

$$\mathbf{U}^{(j+1)} = \mathbf{U}^{(j)} + \epsilon \delta \mathbf{U}^*, \quad (24)$$

where  $\epsilon$  defines the step length based on a simple backtracking line search. The control update step is repeated until a convergence criteria is satisfied. The full algorithm is outlined in Algorithm 1.

#### D. Parameterized trajectory optimization

In the previous section, we described a method to compute a sequence of open-loop control actions over a finite horizon. He et al. [23] proposed planning over macro actions (a collection of sequential actions) to reduce planning complexity, and allow planning over a longer time horizon. Planning over path parameters, similar to macro actions, has also been studied by Sim et al. [27]. Here, we apply an optimization over parameterized paths for planning.

For AUV field trials, we desire simple, practical trajectories in order to easily monitor vehicle health and progress. To address this, we assume that each vehicle employs a feedback path-following controller and optimize over parameters that define a trajectory. For example, a center position and radius define a circular path. The control is then a (known) function of the estimated vehicle state and path parameter vector  $\boldsymbol{\theta}$ . The state transition model (1) can be expressed given the controller and parameter vector as

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k(\hat{\mathbf{x}}_k, \boldsymbol{\theta}), \mathbf{w}_k). \quad (25)$$

We iteratively update locally optimal path parameters in the same way control actions were computed. In this case, Jacobians are computed with respect to the path parameters instead of control actions. Finally, we penalize parameter weights (instead of control) within the cost function to ensure a positive definite Hessian. The number of parameters to specify a path is, in general, much smaller than the number of control actions. Therefore, there is also significant computational savings in optimizing over path parameters.

## IV. PLANNING FOR COOPERATIVE LOCALIZATION

Here, we detail the instantiation of the belief space planning algorithm developed in §III to plan a server trajectory to localize a client vehicle. We require that the nominal client trajectory and path-following controller are available at planning time. We further assume that the client vehicle is able to reconstruct the centralized server-client estimator and that communication latency is negligible.

The state space system is represented by the stacked server and client vehicle states  $\mathbf{x}_k = [\mathbf{x}_{s_k}^\top, \mathbf{x}_{c_k}^\top]^\top$ . The state space dynamics of each vehicle are independent and written

$$\begin{bmatrix} \mathbf{x}_{s_{k+1}} \\ \mathbf{x}_{c_{k+1}} \end{bmatrix} = \begin{bmatrix} f_s(\mathbf{x}_{s_k}, \mathbf{u}_{s_k}, \mathbf{w}_{s_k}) \\ f_c(\mathbf{x}_{c_k}, \mathbf{u}_{c_k}, \mathbf{w}_{c_k}) \end{bmatrix}, \quad (26)$$

where  $\mathbf{u}_{s_k}$  represents the only decision variable control action within the planning problem since  $\mathbf{u}_{c_k}$  is computed by the client's path-following controller and a given mission plan.

Relative range constraints constitute a nonlinear observation over the server-client state

$$z_{r_k} = \|\mathbf{x}_{s_k} - \mathbf{x}_{c_k}\|_2 + w_{r_k}, \quad (27)$$

where  $w_{r_k} \sim \mathcal{N}(0, \sigma_k^2)$ . In general, range observations will be a 3D slant range measurement. Within a linear estimator, such as the EKF, range observations add information in one direction: along the vector between the server and client vehicle positions. For underwater navigation, depth is usually well instrumented, so we seek to minimize horizontal position errors. Therefore, the range measurement utility is highest when the server and client vehicles are far apart, i.e., the vector of added information is approximately parallel to the horizontal plane.

In all simulations presented in this paper, we use a cost function that penalizes control action and client vehicle uncertainty. Unlike many planning problems, there is no desired terminal state for the server. We can define a quadratic stage cost penalizing client uncertainty fitting the form of the objective in (4)

$$\begin{aligned} \text{tr}(\Sigma_{c_k}) &= \mathbf{m}^\top \mathbf{b}_k \\ \text{tr}^2(\Sigma_{c_k}) &= \mathbf{b}_k^\top \underbrace{\mathbf{m} \mathbf{m}^\top}_{\mathbf{Q}_k} \mathbf{b}_k, \end{aligned} \quad (28)$$

where  $\mathbf{m}^\top$  is a row vector that sums the diagonal elements of the covariance matrix corresponding to the client vehicle from the belief state. We simply penalize the control action with  $\mathbf{R}_k = \mathbf{I}$ .

## V. NUMERICAL SIMULATIONS

We first present an empirical channel model informed by reception statistics collected during AUV field trials. We then validate our planning framework through several simulated scenarios where we employ this channel model for planning.

### A. Empirical channel model

In this section, we review how we compute the distribution over channel states  $p(\gamma_k = 1) = \lambda_k$ . Physics-based and empirical models exist for studying communication error in underwater channels. One such empirical channel model proposed by Stojanovic [28] was used by Hollinger et al. [29] for studying multiple vehicle coordination. Within the model, the probability of successful transmission is written as a function of the transmit frequency, power, and environmental conditions. In this work, we are interested in showing how a channel model can be integrated into the planning framework. As such, we employed a simplified model where

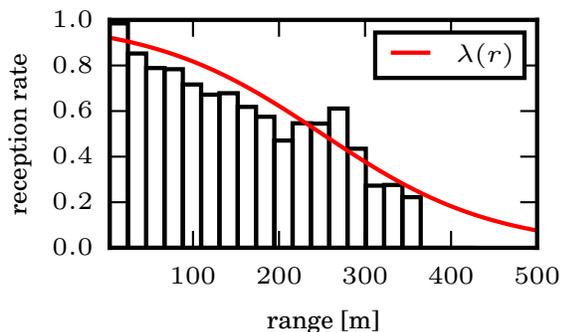


Fig. 2: Reception rate data and channel state model. The histogram represents the frequency of successfully received acoustic broadcasts. The channel model,  $\lambda$ , illustrates (29) tuned to the environment.

the probability of successful reception falls off as a sigmoid with the horizontal range,  $r = \|\mathbf{x}_s - \mathbf{x}_c\|_2$ , between server and client

$$\begin{aligned} \lambda_k(r) &= p(\gamma_k = 1) \\ &= \left[ 1 + \exp \left\{ \frac{1}{\tau} (r - \bar{r}) \right\} \right]^{-1}, \end{aligned} \quad (29)$$

where  $\bar{r}$  is the range at which reception probability is  $\frac{1}{2}$  and  $\tau$  is the length scale that defines the rate at which reception decreases.

We manually fit the parameters of this simple model to data collected over the course of several field trials. We fielded two Iver2 AUVs and a surface ship. Each vehicle was equipped with a Woods Hole Oceanographic Institution (WHOI) Micro-modem and co-processor board [30], 25 kHz BTech Acoustics 2RCL transducer, and a synchronous-clock reference. Each vehicle periodically transmitted Micro-modem Rate 1 and 2 data packets which each consist of three 64 B phase-shift keying (PSK) encoded data frames.

Over the course of three trials, we collected over 3000 range observations. The trials were conducted in shallow water (roughly 20 m) with all transducers suspended at approximately 10 m. While the true inter-vehicle range is unknown, we estimated range from the filtered online vehicle position estimates in post process. The inter-vehicle range did not exceed 400 m. Fig. 2 shows the channel model (29) overlaid on a histogram of reception frequencies for varying relative range, where the channel model matches the characteristic behavior of the true channel.

### B. Simulation: static beacon localization

Our first set of simulations helps illustrate the planning problem for a simple scenario—compute a server trajectory to localize a static beacon. This is a special case of server/client localization in which the client vehicle is fixed, and, hence, does not accrue any additional uncertainty over time. This problem is akin to surveying long-baseline (LBL) network beacons. Jakuba et al. [31] recommend that the topside ship circle the beacons at a fixed radius (up to water depth), providing a heuristic to balance position information and communication. Below, we vary relative depth and channel parameters to demonstrate that the planner

adapts to different conditions, balancing communication and navigation utility.

We assume that the server vehicle follows a simple unicycle model. The server translates a fixed distance,  $\Delta$ , for each timestep with control defined by a change in heading,  $u_k$ , perturbed by zero-mean independent Gaussian noise  $w_k$ . The server state is parameterized by its  $(x, y)$  horizontal position and heading  $\theta$

$$\begin{bmatrix} x_{s_{k+1}} \\ y_{s_{k+1}} \\ \theta_{s_{k+1}} \end{bmatrix} = \begin{bmatrix} x_{s_k} + \Delta \cos(\theta_{s_k} + u_k + w_k) \\ y_{s_k} + \Delta \sin(\theta_{s_k} + u_k + w_k) \\ \theta_{s_k} + u_k + w_k \end{bmatrix}.$$

The beacon (client) state is simply its  $(x, y)$  position. Following each motion step, the server vehicle observes its own position and orientation and the range to the beacon (depending on the channel state).

We optimized over open-loop control actions (‘control’ in Fig. 3) and parameterized circular paths (‘circle’ in Fig. 3). For the open-loop control trajectories, the server vehicle was initialized in the same position for each trial with constant fin angle producing a partial circle around the beacon (although a straight line initialization also converged to circular trajectories). For the parametric circular trajectories, the initial radius was set to 50 m. Fig. 3 illustrates planned server trajectories for varying channel model and relative depth parameters over 50 planning steps. The top row in Fig. 3 demonstrates that our planning framework responds to varying channel parameters. The bottom row shows that information added by each range measurement changes as a function of relative depth between the server and beacon. As the relative depth increases, the server must move farther away in order to localize the beacon. The run times of our non-optimized Python code are listed below each figure.

These simulations also help to illustrate the utility of parameterized trajectories, provided that the path type is well informed by experience. For this simplified scenario, the planned control and parameterized circle paths do not differ significantly and agree with heuristic LBL localization methods, i.e., that circular trajectories provide efficient localization.

### C. Simulation: server/client localization

In the second set of simulated trials, we consider two mobile AUVs—a server and client—and show that the planning framework scales to realistic mission profiles. Each vehicle’s state is parameterized by its  $(x, y)$  horizontal position and heading  $\theta$ . We assume a constant forward velocity so that over a fixed time step, the change in position is  $\Delta$ . The state dynamics follow a simplified bicycle model with steering angle control input  $u_k$

$$\begin{bmatrix} x_{s_{k+1}} \\ y_{s_{k+1}} \\ \theta_{s_{k+1}} \end{bmatrix} = \begin{bmatrix} x_{s_k} + (\Delta + w_{\Delta_k}) \cos(\theta_{s_k}) \\ y_{s_k} + (\Delta + w_{\Delta_k}) \sin(\theta_{s_k}) \\ \theta_{s_k} + \alpha(\Delta + w_{\Delta_k}) \tan(u_k + w_{u_k}) \end{bmatrix},$$

where  $\alpha$  is the reciprocal ‘wheel-base’ length and  $w_{\Delta_k}$  and  $w_{u_k}$  represent Gaussian noise perturbations in the step length and control, respectively.

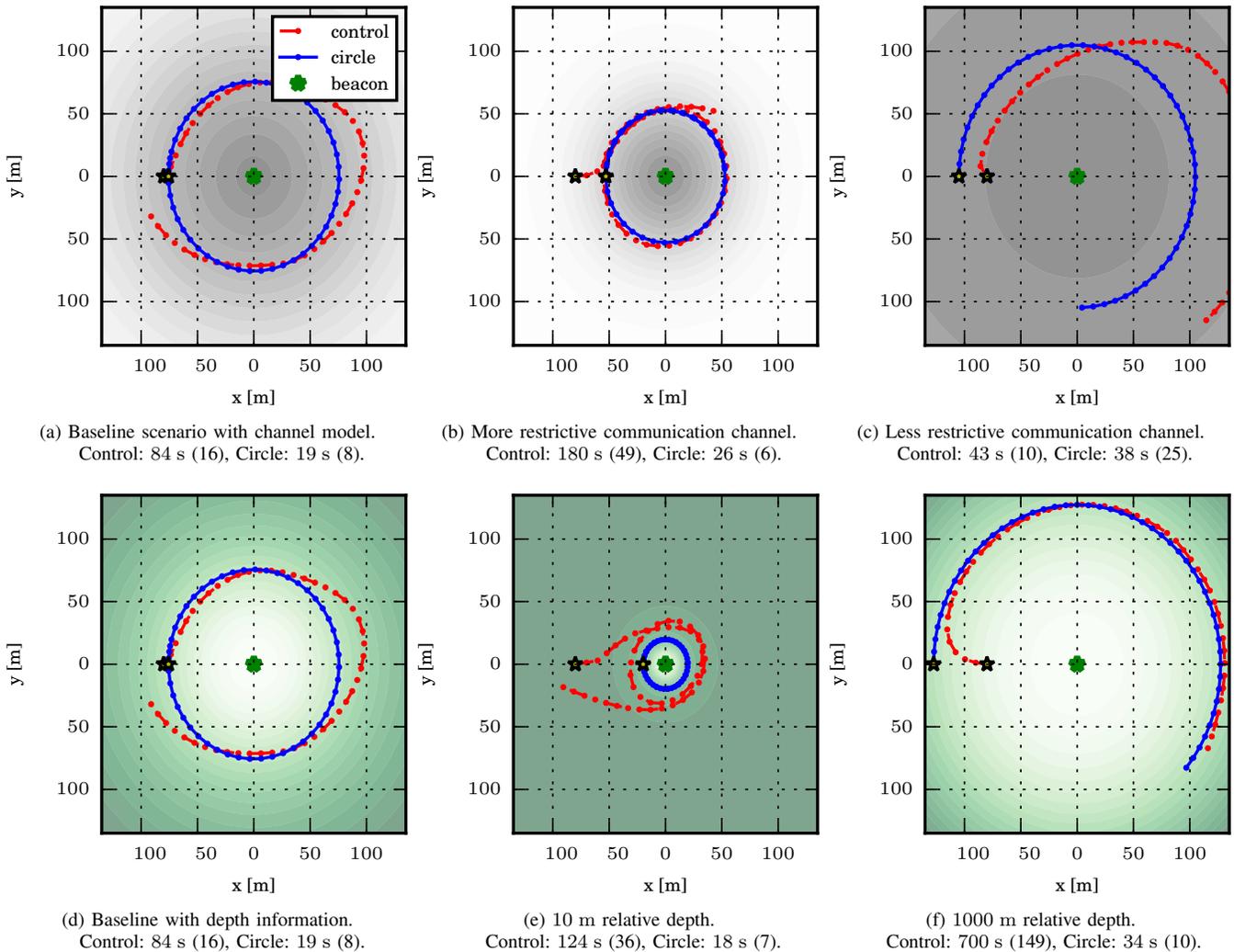


Fig. 3: Example planning for static beacon localization. Top row shows planned trajectories with the underlying channel model using the baseline relative depth (color gradient in top row represents communication channel quality, darker is better). Bottom row plots range information using the baseline channel model (color gradient in bottom row represents range information, darker is better). Planning run times are listed below each figure with number of iterations in parentheses.

We are provided the client survey trajectory at planning time. The estimated survey time is 76 min and is represented by roughly 5000 state transitions. The relative vehicle depth is 100 m and the channel model parameters are  $\bar{r} = 250$  m and  $\tau = 125$  m. By optimizing over ‘diamond’ path parameters (center position, width, height), we reduce the complexity of the problem from several thousand control actions to just four path parameters. Each vehicle relies on a pure-pursuit path-following controller. We also compared the optimized path to a naïvely planned path, which we have employed in several previous cooperative localization field trials. We set the parameters of the naïve path to be centered on the client survey area and span the length and width of the survey area. The naïve path parameters were used to initialize the optimization.

Fig. 4 depicts the planned trajectory for each vehicle. Table I summarizes the performance of each path averaged over 100 simulated trials. Overall, the optimized path receives more OWTTs per trial than the naïvely planned trajectory. Moreover, the client uncertainty (indicated by the

TABLE I: Average performance comparing an optimized path and a naïvely planned path over 100 simulated trials.

METHOD	OWTTs RECEIVED	$ \Sigma_{c_k} $	$\text{tr}(\Sigma_{c_k})$
Proposed (optimized)	67.6	0.40	69.56
Naïvely planned	62.9	1.60	402.34

determinant and trace in the table) is far lower for the optimized path. The optimized server path remains closer on average to the client, and thus is able to more reliably make OWTT observations.

## VI. CONCLUSION

Cooperative underwater localization enables teams of vehicles to exploit range observations to improve their navigation estimates. The quality of the estimate, however, depends on the relative position of the vehicles. Therefore, planning relative paths can benefit online navigation performance. We have presented a probabilistic channel model that represents randomness in observation events. The channel model holds potential utility in other applications involving unknown

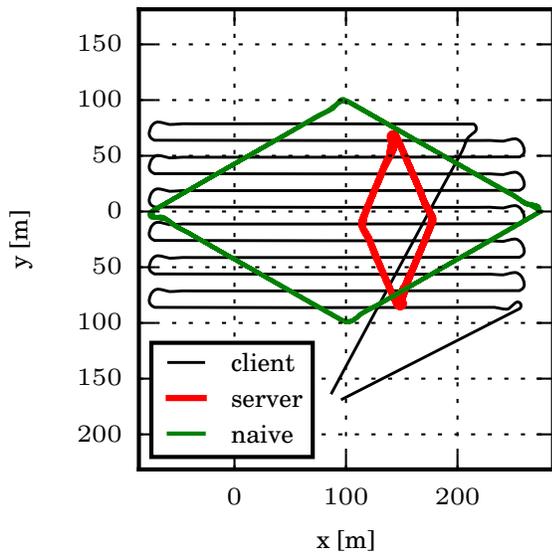


Fig. 4: Optimized (red) and naïvely planned (green) server trajectories.

measurement acquisition. We then integrated the model into a belief space planning framework that optimizes either open-loop control actions or path parameters in order to produce practical AUV trajectories.

#### REFERENCES

- [1] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "Underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conf. Manoeuvring and Cont. Marine Craft*, Lisbon, Portugal, Sep. 2006.
- [2] M. F. Fallon, G. Papadopoulos, J. J. Leonard, and N. M. Patrikalakis, "Cooperative AUV navigation using a single maneuvering surface craft," *Int. J. Robot. Res.*, vol. 29, no. 12, pp. 1461–1474, 2010.
- [3] R. M. Eustice, H. Singh, and L. L. Whitcomb, "Synchronous-clock one-way-travel-time acoustic navigation for underwater vehicles," *J. Field Robot.*, vol. 28, no. 1, pp. 121–136, 2011.
- [4] L. Paull, G. Huang, M. Seto, and J. J. Leonard, "Communication-constrained multi-AUV cooperative SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, Seattle, WA, May 2015, pp. 509–516.
- [5] J. M. Walls, A. G. Cunningham, and R. M. Eustice, "Cooperative localization by factor composition over a faulty low-bandwidth communication channel," in *Proc. IEEE Int. Conf. Robot. and Automation*, Seattle, WA, May 2015, pp. 401–408.
- [6] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, no. 1–2, pp. 99–134, 1998.
- [7] J. M. Porta, M. T. J. Spaan, and N. Vlassis, "Robot planning in partially observable continuous domains," in *Proc. Robot.: Sci. & Syst. Conf.*, Cambridge, USA, Jun. 2005.
- [8] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11–12, pp. 1448–1465, 2009.
- [9] A. Agha-mohammadi, S. Chakravorty, and N. M. Amato, "FIRM: Sampling-based feedback motion-planning under motion uncertainty and imperfect measurements," *Int. J. Robot. Res.*, vol. 33, no. 2, pp. 268–304, 2014.
- [10] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. and Automation*, Shanghai, China, May 2011, pp. 723–730.
- [11] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez, "Belief space planning assuming maximum likelihood observations," in *Proc. Robot.: Sci. & Syst. Conf.*, Zaragoza, Spain, Jun. 2010.
- [12] J. van den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *Int. J. Robot. Res.*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [13] S. Patil, G. Kahn, M. Laskey, J. Schulman, K. Goldberg, and P. Abbeel, "Scaling up Gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation," in *Proc. Int. Workshop on the Algorithmic Foundations of Robotics*, Istanbul, Turkey, Aug. 2014.
- [14] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. Robot.: Sci. & Syst. Conf.*, Cambridge, MA, Jun. 2005.
- [15] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, Barcelona, Spain, Apr. 2005, pp. 661–666.
- [16] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *Int. J. Robot. Res.*, 2015, in press.
- [17] V. Kumar, N. Leonard, and A. S. Morse, *Cooperative Control*, ser. Lecture Notes in Control and Information Sciences. Springer Verlag, 2004, vol. 309.
- [18] S. Martinez and F. Bullo, "Optimal sensor placement and motion coordination for target tracking," *Automatica*, vol. 42, no. 4, pp. 661–668, 2006.
- [19] B. Charrow, N. Michael, and V. Kumar, "Cooperative multi-robot estimation and control for radio source localization," *Int. J. Robot. Res.*, vol. 33, no. 4, pp. 569–580, 2014.
- [20] C. R. German, M. V. Jakuba, J. C. Kinsey, J. Partan, S. Suman, A. Belani, and D. R. Yoerger, "A long term vision for long-range ship-free deep ocean operations: Persistent presence through coordination of autonomous surface vehicles and autonomous underwater vehicles," in *Proc. IEEE/OES Autonomous Underwater Vehicles Conf.*, Southampton, UK, Sep. 2012, pp. 1–7.
- [21] Y. T. Tan, R. Gao, and M. Chitre, "Cooperative path planning for range-only localization using a single moving beacon," *IEEE J. Ocean. Eng.*, vol. 39, no. 2, pp. 371–385, 2014.
- [22] A. Bahr, J. J. Leonard, and A. Martinoli, "Dynamic positioning of beacon vehicles for cooperative underwater navigation," in *Proc. IEEE/RSSJ Int. Conf. Intell. Robots and Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 3760–3767.
- [23] R. He, E. Brunskill, and N. Roy, "Efficient planning under uncertainty with macro-actions," *J. Artif. Intell. Res.*, vol. 40, pp. 523–570, 2011.
- [24] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, 2004.
- [25] A. Kim and R. M. Eustice, "Perception-driven navigation: Active visual SLAM for robotic area coverage," in *Proc. IEEE Int. Conf. Robot. and Automation*, Karlsruhe, Germany, May 2013, pp. 3196–3203.
- [26] M. Diehl, "Numerical optimal control," 2011. [Online]. Available: <http://homes.esat.kuleuven.be/~mdiehl/TRENTO/numopticon.pdf>
- [27] R. Sim, G. Dudek, and N. Roy, "Online control policy optimization for minimizing map uncertainty during exploration," in *Proc. IEEE Int. Conf. Robot. and Automation*, vol. 2, New Orleans, LA, Apr. 2004, pp. 1758–1763.
- [28] M. Stojanovic, "On the relationship between capacity and distance in an underwater acoustic communication channel," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 11, no. 4, pp. 34–43, 2007.
- [29] G. Hollinger, S. Yerramalli, S. Singh, U. Mitra, and G. Sukhatme, "Distributed data fusion for multirobot search," *IEEE Trans. Robot.*, vol. 31, no. 1, pp. 55–66, 2015.
- [30] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The WHOI micro-modem: An acoustic communications and navigation system for multiple platforms," in *Proc. IEEE/MTS OCEANS Conf. Exhib.*, Washington, D.C., Sep. 2005, pp. 1086–1092.
- [31] M. V. Jakuba, C. N. Roman, H. Singh, C. Murphy, C. Kunz, C. Willis, T. Sato, and R. A. Sohn, "Field report: Long-baseline acoustic navigation for under-ice AUV operations," *J. Field Robot.*, vol. 25, no. 11–12, pp. 861–879, 2008.